

MPEG-2 High-Definition Decoder on the DM642 EVM

Video and Imaging Systems

ABSTRACT

The software demonstrates the MPEG-2 high-definition (HD) decoder running on the DM642 Evaluation Module (EVM). The demonstration uses the MPEG-2 HD decoder to decode the MPEG-2 HD bitstream and display the decoded frames on the HD output device.

The demonstration uses:

- MPEG-2 decoder library optimized for a DM642 EVM
- MPEG-2 decoder library implemented using XDAIS interfaces
- Sample integration of the MPEG-2 decoder library using RF-5 framework

Contents

Software Architecture/Data Flow	2
Framework Flowchart.....	2
System Requirements/Configuration	4
Software Requirements.....	4
Hardware Requirements	5
Hardware Setup	5
Demonstration Execution	6
Demonstration Code and Build Procedure	6
Build Procedure.....	7
Changing the Input Bitstream.....	7
Known Constraints	8

Figures

Figure 1. Data Flow Diagram	2
Figure 2. Framework Flowchart	4
Figure 3. Hardware Setup	5
Figure 4. Directory Structure for mpeg2_decoder_hd	6

Software Architecture/Data Flow

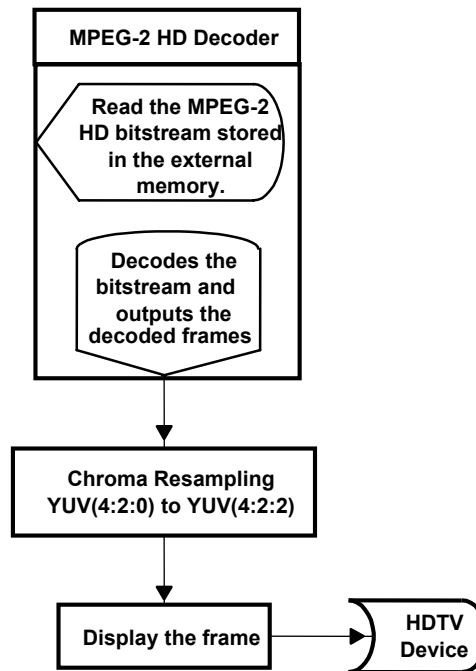


Figure 1. Data Flow Diagram

The data flow in the demonstration follows this sequence:

1. The MPEG-2 test-stream stored in the external memory of the EVM board is read.
2. The read MPEG-2 bitstream is passed to the MPEG-2 decoder module.
3. The MPEG-2 decoder module decodes the MPEG-2 bitstream and outputs the decoded frames.
4. The decoded frames received from the MPEG-2 decoder module are in YUV 4:2:0 format and are resampled to YUV 4:2:2 format.
5. The frame is then displayed on the HD output device (HDTV).

Framework Flowchart

The demonstration uses RF-5 framework to integrate the MPEG-2 decoder library in the system. The demonstration uses a two-task model. Before coming to the DSP BIOS™ task scheduler, the demonstration code initializes various modules used in the system. These include :

- Board and processor
 - The systems performs DSP BIOS™ initialization and CSL initialization.
 - The L2 cache mode is set to 64K cache.
 - EMIFA CE0 and EMIF CE1 spaces are enabled for caching.

- The DMA priority queue lengths are set to maximum.
- Priority for L2 request is set as high.
- RF- 5 modules
 - The systems initializes the channel module of RF-5.
 - The system initializes ICC and SCOM modules of RF-5 required for intercell communication and messaging.
 - Channel setup is performed with the internal, external, and scratch heap buffers.
- Display channel
 - An instance of display channel is created and started.
- Algorithm instances
 - The MPEG-2 decoder cell is created and registered in the channel.
 - The channel is opened, which leads to creation of the instance of the decoder cell.

After the initializations, the system enters the two-task system managed by the DSP BIOS™ scheduler. These two tasks use the SCOM module of RF-5 to communicate with each other:

- Process task

The process task is responsible for passing the bitstream to the decoder module, decoding the frame, and passing it on to the output task. The process task achieves the MPEG-2 decoding by executing the RF-5 channel, which has the MPEG-2 HD decoder cell registered in it. The ICC module manages the passing of bit-stream buffer to the decoder cell.

The task waits until it receives the message from the output task to continue further decoding. The RF-5 channel in the demonstration code consists of an MPEG-2 HD decoder cell. During the execution of the channel , the bitstream buffer is passed to the decoder cell and the decoder cell executes to produce the decoded frames. It then sends a message to the output task with the output frame pointer embedded in the message. The task then waits for the message from the output task to continue.
- Output task

The output task is responsible for displaying the frames on the HD output device. It uses FVID_exchange calls provided by the driver to display a frame on the HD output device.

The acquired frame is in YUV 4:2:0 format and is resampled to YUV 4:2:2 format. It then sends a message to the process task to continue. The task then waits for the message from the process task to continue.

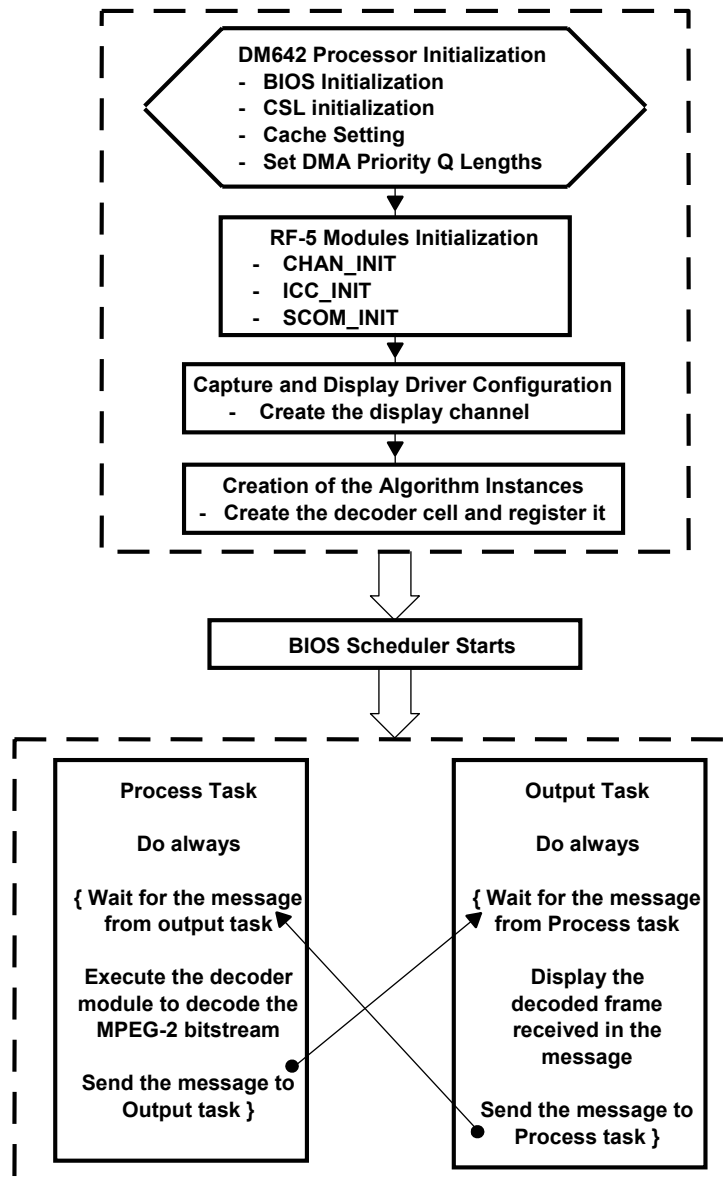


Figure 2. Framework Flowchart

System Requirements/Configuration

Software Requirements

- Microsoft Windows NT (SP6)/Microsoft Windows 2000 (SP1 and SP2)
- Code Composer Studio™ Integrated Development Environment (IDE) version 2.20.18
- Driver software (DDK 1.1)

Hardware Requirements

- Pentium machines with 450 MHz, 64MB RAM (minimum)
- DM642 EVM

NOTE

The DM642 EVM requires a few small modifications on the board to work properly for the HD display. Refer to the TMS320DM642 Evaluation Module Technical Reference (Spectrum Digital Document Number: 506845-0001) for more details about the required modification.

- HDTV for display purposes
- XDS 510/560 emulator

Hardware Setup

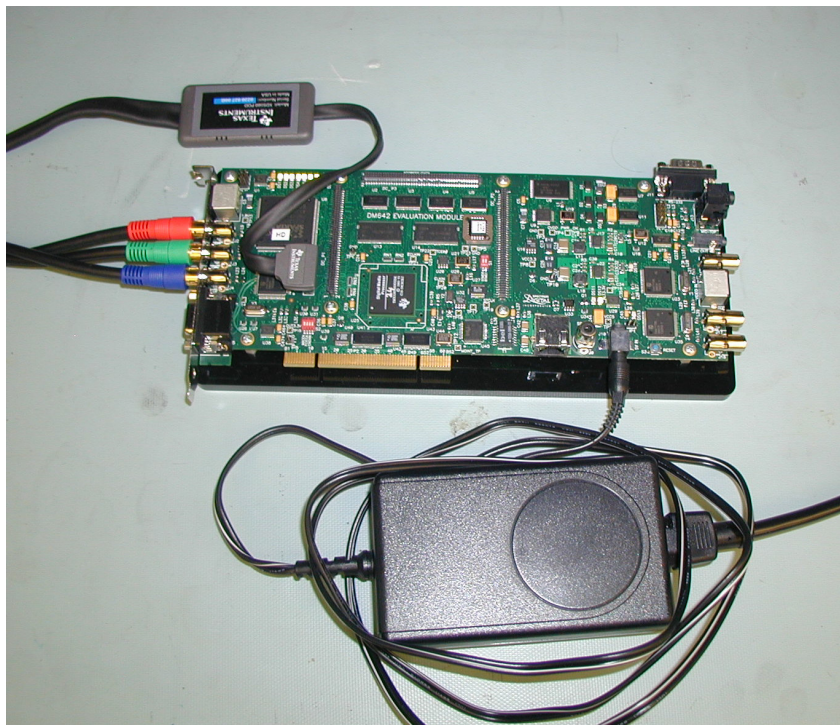


Figure 3. Hardware Setup

To run the demonstration, the hardware must be set up properly, as shown in Figure 3.

- The XDS510/560 emulator must be connected to JTAG pins to download the demonstration code to the board and control it from Code Composer Studio™ IDE.
- The three RCA jacks for component outputs Y, Pr, and Pb must be connected to the respective component inputs of the HD output device (HDTV). Refer to the technical

reference guide (Spectrum Digital Document Number 506845-0001) for more information on RCA jacks and their location (J2: Red, Pr; J3: Green,Y; and J4:Blue, Pb).

- The DM642 EVM must be connected to the appropriate power source.

Demonstration Execution

To run the demonstration:

1. Set up the hardware as described.
2. Power up the DM642 EVM board.
3. Start Code Composer Studio™ IDE version 2.20.18.
4. Check the color bar on the output device.
5. Go to the bin folder under the mpeg2_decoder_hd directory and load .out.
6. Once the program is loaded, go to the Debug Menu and press the Run option (F5).
7. On the HD output device screen, watch the frames decoded from the MPEG-2 HD bitstream, with the TI logo on the top-right corner of the frames.

Demonstration Code and Build Procedure

The demonstration code for MPEG-2 HD decoder is located in `evmdm642\examples\video\MPEG2_decoder_hd` directory.

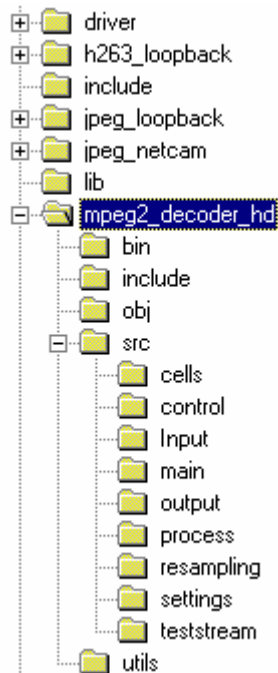


Figure 4. Directory Structure for mpeg2_decoder_hd

Build Procedure

1. Start Code Composer Studio™ IDE version 2.20.18.
2. Open the MPEG-2 HD decoder project (mpeg2_hd_decoder_dm642.pjt) from examples\video\MPEG2_Decoder_HD folder.
3. Go to Project->Build Options->Compiler->Preprocessor and define the symbols as required for appropriate demonstration setup.

Default options: The following must always be defined for proper compilation of the demonstration: CHIP_DM642, C6000, and HDTV.
4. Define UTL_DBGLEVEL= 60, if the monitoring of statistics is desired using DSP BIOS™ STS objects.
5. Define SHOW_TIME, to log the cycles consumed for decoding in the LOG_timer log object.
6. If the C_DIR is not defined or the DDK package has been installed outside the Code Composer Studio™ IDE folder, modify the include paths to point to the appropriate <ccs install dir>\ti\... paths in the project build options. If the C_DIR is defined properly, there is no need to modify the include paths.
7. Build the project and load the executable mpeg2_hd_decoder.out build in the examples\video\mpeg2_decoder_hd\bin directory.
8. Before running the executable, make sure the output (HDTV) is connected correctly.
9. Press F5 to watch the frames decoded from the MPEG-2 HD bitstream, with the TI logo at the top-right corner of the frames.

Changing the Input Bitstream

The demonstration project contains an obj file (city.obj) for the MPEG-2 HD bitstream. The test-stream in the project can be changed to any other bitstream by adding the .obj file or .asm file for another bitstream. A utility raw2asm.exe can be found in the utils folder that converts any MPEG-2 test-stream file to a corresponding .asm file.

The raw2asm utility can be used as follows:

Usage: raw2asm <raw bit stream file> <asm file> <buffer name> <section_name>

Here buffer name is the array name for storing the whole bitstream, and section name is the data section in which this array should reside. This section name is used in the .cmd file to properly place this array in the desired memory area.

This project expects the buffer name of the bitstream to be share_bsbuf_storage, and the section for the same is dram10. Therefore, use the utility as follows:

raw2asm <raw bitstream file> <asm file> share_bsbuf_storage dram10

The .asm file generated by this utility can be added directly to the project or the .obj can be made from the .asm file using the C6x compiler (cl6x.exe).

Known Constraints

The MPEG2 library in the project has been compiled to work in conformance with the main profile @ high level (MP@HL), as suggested in the ISO/IEC document 13818-2:1995.

References

1. TMS320DM642 Evaluation Module Technical Reference (Spectrum Digital Document Number: 506845-0001)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2003, Texas Instruments Incorporated